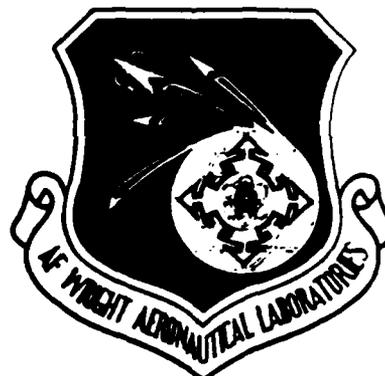


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FILE COPY

2

AFWAL-TR-86-4006
Volume VIII
Part 30



AD-A182 660

INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 30 - Text Editor User Manual

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

Final Report for Period 22 September 1980 - 31 July 1985
November 1985

Approved for public release; distribution is unlimited.

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

DTIC
JUL 27 1987
E

87 7 22 058

Unclassified

A182 66C

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFVAL-TR-86-4006 Vol VIII, Part 30		
6a. NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFVAL/MLTC		
6c. ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345		7b. ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6533		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF	8b. OFFICE SYMBOL (If applicable) AFVAL/MLTC	8. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-80-C-5155		
8c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433		10. SOURCE OF FUNDING NOS		
		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 7500	TASK NO. 62
11. TITLE (Include Security Classification) (See Reverse)				
12. PERSONAL AUTHOR(S) Morenc, Carol and Stafford, Frances				
13a. TYPE OF REPORT Final Technical Report	13b. TIME COVERED 22 Sept 1980 - 31 July 1985	14. DATE OF REPORT (Yr, Mo, Day) 1985 November	15. PAGE COUNT 45	
16. SUPPLEMENTARY NOTATION ICAM Project Priority 6201		The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.		
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP			SUB GR
1508	0905			
19. ABSTRACT (Continue on reverse if necessary and identify by block number)				
<p>7 The Text Editor is an application in the IISS that provides users with file editing capabilities. Editing functions include inserting, deleting, moving, and replacing text.</p> <p>The Text Editor operates in two modes, edit and command. Edit mode is used to enter text and perform editing functions by pressing function keys. Command mode is used to perform edition functions that require additional information from the user such as a file name or search string.</p>				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson		22b. TELEPHONE NUMBER (Include Area Code) 613-255-6976	22c. OFFICE SYMBOL AFVAL/MLTC	

11. Title

Integrated Information Support System (IISS)
Vol VIII - User Interface Subsystem
Part 30 - Text Editor User Manual

A S D 86 0043
9 Jan 1986

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



PREFACE

This user's manual covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors

Role

Boeing Military Aircraft
Company (BMAC)

Reviewer.

D. Appleton Company
(DACOM)

Responsible for IDEF support,
state-of-the-art literature
search.

General Dynamics/
Ft. Worth

Responsible for factory view
function and information
models.

<u>Subcontractors</u>	<u>Role</u>
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business.
North American Rockwell	Reviewer.
Northrop Corporation	Responsible for factory view function and information models.
Pritsker and Associates	Responsible for IDEF2 support.
SofTech	Responsible for IDEF0 support.

TASKS 4.3 - 4.9 (TEST BED)

<u>Subcontractors</u>	<u>Role</u>
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

Subcontractors

Role

Digital Equipment Corporation (DEC)

Consulting and support of the performance testing and on DEC software and computer systems operation.

McDonnell Douglas Automation Company (McAuto)

Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.

On-Line Software International (OSI)

Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.

Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)

Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.

SofTech, Inc.

Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.

Software Performance Engineering (SPE)

Responsible for directing the work on performance evaluation and analysis.

Structural Dynamics Research Corporation (SDRC)

Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC).

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP).
D. Appleton Company (DAGOM)	1502	IISS enhancements to Integration Methodology.
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements.
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI).
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

	<u>Page</u>
SECTION 1.0 INTRODUCTION	1-1
1.1 Functional Summary	1-1
1.2 Intended Audience	1-1
1.3 Documentation Conventions	1-1
SECTION 2.0 DOCUMENTS	2-1
2.1 Reference Documents	2-1
2.2 Terms and Abbreviations	2-2
SECTION 3.0 FILE EDITING	3-1
3.1 Starting an Editing Session	3-1
3.2 Insert Line	3-2
3.3 Entering Text	3-3
3.4 Midline Break	3-3
3.5 Delete a Line of Text	3-4
3.6 Viewing the Workspace Buffer	3-6
3.6.1 First Page	3-6
3.6.2 Last Page	3-7
3.7 Selecting and Moving Text	3-8
3.7.1 Selecting Text to be Moved	3-8
3.7.2 Inserting Text from the Cut and Paste Buffer	3-10
3.7.2.1 Using the Paste Function to Insert Text	3-10
3.7.2.2 Using the Fill Function to Insert Text	3-11
3.8 Quitting	3-12
3.9 Executing Command Mode Commands	3-13
3.10 Searching for Text	3-15
3.10.1 SEARCH Operands	3-15
3.10.2 Finding the Next Occurrence	3-16
3.10.3 Considerations	3-16
3.11 Replacing One Text String with Another	3-17
3.11.1 Specifying the Scope	3-17
3.11.2 REPLACE Operands	3-17
3.11.3 Considerations	3-18
3.12 Loading a File for Editing	3-19
3.12.1 LOAD Operand	3-19
3.12.2 Considerations	3-20
3.13 Saving the Workspace Buffer	3-20
3.13.1 SAVE Operand	3-20

3.13.2	Considerations	3-21
3.14	Clearing the Workspace Buffer	3-21
3.15	Setting the Fill Margins	3-22
3.15.1	MARGINS Operands	3-22
3.15.2	Considerations	3-23
3.15.3	Example Showing Affects of Setting Fill Margins	3-23
3.16	Repeating a Command or Function Key	3-24
3.16.1	REPEAT Operands	3-25
3.16.2	Considerations	3-25
3.17	Naming the Workspace Buffer	3-26
3.17.1	BUFFER Operand	3-26

APPENDICES

APPENDIX A	FORM PROCESSOR(FP)/TEXT EDITOR(TE) CALLABLE INTERFACE	A-1
------------	--	-----

FIGURES

FIGURE 1-1	Text Editor Application Keypad Layout	1-2
------------	--	-----

SECTION 1

INTRODUCTION

1.1 Functional Summary

The Text Editor (TE) is an application in the IISS that provides users with file editing capabilities. Editing functions include inserting, deleting, moving, and replacing text.

There are two Text Editor modes, edit and command.

- Edit mode is used to enter text and perform editing functions by pressing function keys.
- Command mode is used to perform editing functions that require additional information from the user such as a file name or search string. The command mode commands are SEARCH, REPLACE, LOAD, SAVE, CLEAR, MARGINS, REPEAT, BUFFER.

NOTE: Do not confuse the modes of the Text Editor described above with the IISS keyboard modes such as "application", "scrl/page", "text edit" and "window mgr". The Text Editor documented in this manual is an application available in the IISS and in order to use the special function keys you must set the IISS mode field to "application".

The IISS keyboard modes are documented in section 3 of the Terminal Operator Guide. One of these modes is the Text Edit Mode. It is used to edit items when entering information into forms.

1.2 Intended Audience

This manual is intended for people who want to create or modify files using the Text Editor Application of the IISS.

1.3 Documentation Conventions

This manual assumes that the TE is running in a VAX/VMS environment.

This manual also assumes that the user is familiar with the concept of buffers. Two buffers are used in the TE. They are

the workspace buffer and the cut and paste buffer. When editing an existing file, a duplicate copy of the file's contents is put into the workspace buffer. The contents of this buffer appears on your terminal screen a page at a time. On a VT100 terminal, each page is 22 lines of text. The text that you insert or revise in workspace stays there until you save your edits (with the SAVE command) or delete them (with CLEAR or QUIT). If you save your edits, a new file is created. Existing versions of the file remain unchanged. The cut and paste buffer is used as temporary storage to enable you to move text around in the workspace buffer.

In the examples, the current cursor position is indicated by an underscore. To conserve space, the example screen page size is 10 lines except when more lines are needed to demonstrate the functionality.

Uppercase words enclosed in angle brackets denote function keys (i.e., <QUIT>). Each key will be described in terms of the operation that is performed when you press that key. The actual keyboard or keypad key that you use is terminal dependent and is shown in the terminal mapping charts in the Terminal Operator Guide. Figure 1-1 shows the function key layout on a VT100 keypad. Because the Text Editor is an application of the IISS, the mode item which is in the lower right corner of the screen, must be toggled to display "application" when you want to use the Text Editor function keys. How to change modes and a description of the other available modes is explained in the IISS Terminal Operator Guide.

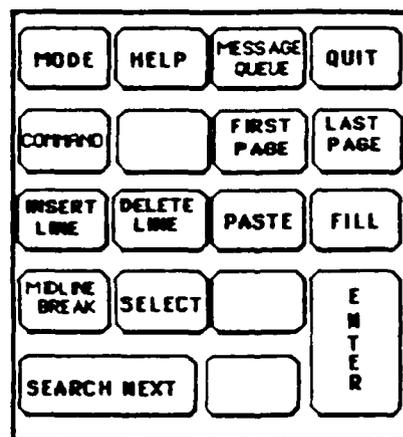


Figure 1-1 Text Editor Application Keypad Layout

SECTION 2
DOCUMENTS

2.1 Reference Documents

- [1] Structural Dynamics Research Corporation, Text Editor Development Specification, DS 620144600C, 1 November 1985.
- [2] Systran, ICAM Documentation Standards, ICAM Document IDS150120000c, 15 September 1983.

This manual is one of a set of user manuals that together describe how to operate in the IISS environment. The complete set consists of the following manuals listed here for reference:

- [1] Structural Dynamics Research Corporation, IISS Form Editor User Manual, UM 6201444400B, 1 November 1985.

Explains how to define and maintain electronic forms. It is intended to be used by programmers writing application programs that use the Form Processor.
- [2] Structural Dynamics Research Corporation, IISS Form Processor User Manual, UM 620144200B, 1 November 1985.

Describes the set of callable execution time routines available to an application program to process electronic forms. It is intended to be used by programmers writing application programs for the IISS environment.
- [3] Structural Dynamics Research Corporation, IISS Terminal Operator Guide, OM 620144000, 1 November 1985.

Explains how to operate the generic IISS terminal when running an IISS application program. The IISS end user environment, function selection and some predefined applications are also described.
- [4] Structural Dynamics Research Corporation, IISS Text Editor User Manual, UM 620144600B, 1 November 1985.

Explains how to use the file editing functions

including: inserting, deleting, moving and replacing text.

- [5] Structural Dynamics Research Corporation, IISS Rapid Application Generator User Manual, UM 620144502 , 1 November 1985.

Describes the Application Definition Language and the process used for translating textual definitions of interactive database applications into programs that access selected data base information resident in the Common Data Model. This information is accessible through the IISS Neutral Data Manipulation Language.

- [6] Structural Dynamics Research Corporation, IISS Report Writer User Manual, UM 620144501 , 1 November 1985.

Describes the Report Definition Language and the process of creating a hard copy report of selected data base information resident in the Common Data Model. This information is accessible through the IISS Neutral Data Manipulation Language.

- [7] Structural Dynamics Research Corporation, IISS Virtual Terminal User Manual, UM 620144300B, 1 November 1985.

Explains the program callable interface to the IISS Virtual Terminal. The callable routines, Virtual Terminal commands and the implementation of additional terminals are described. It is intended for application and system programmers working in the IISS environment.

2.2 Terms and Abbreviations

Buffer Name: the default file in which the buffer will be saved if no file is given on a save command.

Current Cursor Position: the position of the cursor before an edit command or function is issued in the text editor.

Cursor Position: the position of the cursor after any command is issued.

Cut and Paste Buffer: where deleted lines go and the paste and fill edit commands get their data.

Display Start Line: the first line in the buffer to be displayed.

Display Size: the number of lines used in the edit area.

Field: two-dimensional space on a terminal screen.

Field Pointer: indicates the ITEM which contains the current cursor position.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

Form Processor Text Editor: (FPTE), subset of the Form Processor that consists of software modules that provide text editing capabilities to all users of applications that use the Form Processor.

IISS Function Screen: the first screen that is displayed after logon. It allows the user to specify the function he wants to access and the device type and device name on which he is working.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Message: descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or provide other user information.

Message Line: a line on the terminal screen that is used to display messages.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Previous Cursor Position: the position of the cursor when the previous edit command was issued.

Previous Edit Command: the function key pressed before the current one.

Select Line: one terminus of the select range.

Select Mode: when on, certain commands will be executed over the lines in the selected range. The commands are <DELETE LINE> and replace.

Text Editor: (TE), subset of the IISS User Interface that consists of a file editor that is based on the text editing functions built into the Form Processor.

Top of file: the first line of the buffer.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

UM 620144600B
1 November 1985

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

Virtual Terminal: (VT), subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

3.3 Entering Text

Whatever you type on the keyboard during an editing session is inserted directly before the cursor as text. If you continue to enter text past the right margin, the cursor will automatically move to the next line. The line will break exactly at the right edge of the screen. You are responsible for making sure there are blank lines for the cursor to move into. Any text entered on or following the bottom of file mark is lost.

```
+-----+
|This example shows how a line breaks exactly at the right m|
|argin as you are typing.  If you want to move to the next l|
|ine before reaching the right margin and possibly splitting|
| a word, you must press the <TAB> key.  The workspace buffe|
|r contained at least five blank lines to accept this text.  |
| |
|>>>>buf_bof<<<< |
| |
| >> |
|Msg: | application|
+-----+
```

3.4 Midline Break

The <MIDLINE BREAK> key is used to remove all the characters from the cursor position to the end of the line and insert them at the beginning of a new line immediately following the line containing the cursor. The cursor is then positioned at the beginning of this new line. Pressing the <MIDLINE BREAK> key at the beginning of a line is the same as the insert line function.

The next two examples show how your screen appears before and after pressing the <MIDLINE BREAK> key from the cursor position shown.

```
+-----+
|This is a line of text in the workspace buffer. |
|>>>>buf_bof<<<< |
| |
|>> |
|Msg: | application|
+-----+
```

```
+-----+
|This a line of text |
|in the workspace buffer. |
|>>>>buf_bof<<<< |
| |
|>> |
|Msg: | application|
+-----+
```

A midline break is not permitted on the bottom of file mark or any following line. If attempted, the function is aborted and a warning message is issued. The insert line function is part of a midline break. If the insert line fails, the midline break will also be aborted.

A midline break can be undone using the fill function as described in another section of this manual.

3.5 Delete a Line of Text

The (DELETE LINE) key is used to delete the characters from the current cursor position to the end of the line. The cursor is moved to the start of the next line. If the cursor was at the beginning of the line, all lines which follow are scrolled up one line. The TE does not permit the line containing the bottom of file mark or any following lines to be deleted. An attempt to do so is aborted and a warning message is issued.

The next two examples show how your screen appears before and after pressing the <DELETE LINE> key from the cursor position shown.

```
+-----+
|This is a line of text in the workspace buffer.
|>>>>buf_bof<<<<
|
|
|  >>
|Msg:                                     application|
+-----+
```

```
+-----+
|This is a line of text
|>>>>buf_bof<<<<
|
|
|  >>
|Msg:                                     application|
+-----+
```

One of three things happens to the deleted line.

- If the select mode is on, all the lines in the selected range replace the contents of the cut and paste buffer as explained in the section, "Selecting and Moving Text".
- If the previous function was a delete line, the cursor has not been moved and the select mode is off, the line is concatenated to the cut and paste buffer.

- If the previous function was not a delete line or the cursor was moved, and the select mode is off, the line replaces the current contents of the cut and paste buffer.

Since all deleted lines either replace the contents of the cut and paste buffer or are concatenated to it, you can always undo the last delete function by using the paste or fill functions as explained in another section of this manual.

3.6 Viewing the Workspace Buffer

The functions in this section let you view all the data in the workspace buffer when it contains more lines than the size of the display page. When using these functions, you should think of the data moving around behind a stationary screen.

3.6.1 First Page

The `<FIRST PAGE>` key is used to display the first `n` lines of the workspace buffer where `n` is the terminal display size. The cursor is positioned on the first character of the file. The next two examples show how your terminal screen appears before and after pressing the `<FIRST PAGE>` key.

```
+-----+
|Line 34
|Line 35
|Line 36
|Line 37
|Line 38
|Line 39
|Line 40
|Line 41
|Line 42<FIRST PAGE>
| >>
|Msg:                                     application|
+-----+
```

```
+-----+
|Line 1|
|Line 2|
|Line 3|
|Line 4|
|Line 5|
|Line 6|
|Line 7|
|Line 8|
|Line 9|
| >> |
|Msg:  1 Top of File                application|
+-----+
```

3.6.2 Last Page

The <LAST PAGE> key is used to display the last n lines of the workspace buffer where n is the terminal display size. The cursor is positioned at the bottom of file mark. The next two examples show how your screen appears before and after pressing the <LAST PAGE> key. The first and last page functions are only defined within the context of the Text Editor.

```
+-----+
|Line 9|
|Line 10|
|Line 11|
|Line 12|
|Line 13|
|Line 14|
|Line 15|
|Line 16|
|Line 17<LAST PAGE>|
| >> |
|Msg:                                application|
+-----+
```

```
+-----+
|Line 37
|Line 38
|Line 39
|Line 40
|Line 41
|Line 42
|Line 43
|>>>buf_bof<<<<
|
| >>
|Msg:  1 Bottom of File                               application|
+-----+
```

To review the other pages or portions of your file, you use the scrolling and paging keyboard functions. You access these by putting the terminal keyboard into the IISS "scrl/page" mode as explained in Section 3 of the Terminal Operator Guide.

3.7 Selecting and Moving Text

This section describes how to move text about in the workspace buffer.

3.7.1 Selecting Text to Be Moved

There are two ways to select the text you want to move. One way is to move the cursor to the first line to be moved and press the <DELETE LINE> key for the n consecutive lines to be moved. An alternate way, which is more convenient when selecting large blocks of text, is to go into select mode by pressing the <SELECT> key. From this point you have the option of moving the cursor forward to just after the last character in the text to be moved or backward to the first character in the text to be moved and then pressing the <DELETE LINE> key. In either case, all text between the select line marker and the cursor is stored in the cut and paste buffer.

When the <SELECT> key is pressed, you are in select mode and the message ">>>>>SELECT LINE<<<<<<<" is inserted on the screen. The next two examples show how your screen appears before and after pressing the <SELECT> key from the cursor position shown.

```
+-----+
|Line 1
|Line 2
|Line 3
|Line 4
|Line 5
|>>>>buf_bof<<<<
|
|
|>>
|Msg:                                     application|
+-----+
```

```
+-----+
|Line 1
|Line 2
|Line 3
|>>>>>SELECT LINE<<<<<<<<
|Line 4
|Line 5
|>>>>buf_bof<<<<
|
|
|>>
|Msg:  1 Select Active                       application|
+-----+
```

If you move the cursor backward, the select line marks the end of the select range and the current cursor position is the beginning. If you move the cursor forward, the select line marks the beginning of the select range and the select range will end just before the current cursor position.

The select mode may be cancelled by pressing the <SELECT> key again without pressing the <DELETE LINE> key. Select mode may be entered in the middle of a line. However, it may not be entered at the bottom of file mark or any following lines. If you attempt this, the function will be aborted and a warning message will be issued. The select function uses the insert line function to display the select line message. If the insert line fails, the select is aborted.

3.7.2 Inserting Text from the Cut and Paste Buffer

To insert text from the cut and paste buffer, move the cursor to the line where the text is to be positioned and press the <PASTE> key or the <FILL> key. The contents of the paste buffer will be inserted just before the line containing the current cursor position. Note that the position on the screen of the lines below the current cursor position remain unchanged. This means that all or part of the lines above the cursor and/or the text inserted may be scrolled up. Also note that the lines of text remain in the cut and paste buffer until the <DELETE LINE> key is pressed again so that the lines may be inserted in as many places as required.

3.7.2.1 Using the Paste Function to Insert Text

The paste function restores the deleted lines of text in the exact format they were in when deleted. The next two examples show how your screen appears before and after pressing the <PASTE> key from the cursor position shown. The cut and paste buffer contains the line "This line is in the cut and paste buffer".

```
+-----+
|Line 1|
|Line 2|
|Line 3|
|Line 4|
|Line 5|
|>>>buf_bof<<<<|
|                |
|                |
|>>                |
|Msg:                |application|
+-----+
```

```

+-----+
|Line 2
|Line 3
|Line 4
|This line is in the cut and paste buffer
|Line 5
|>>>>buf_bof<<<<
|
|
|>>
|Msg:
|
+-----+

```

application

3.7.2.2 Using the Fill Function to Insert Text

The fill function restores the deleted text by reformatting the contents of the cut and paste buffer so that each new line is filled to the current fill margins with as many words as possible from the paste buffer. A word is a sequence of nonblank printable characters. The MARGINS command allows you to define the margins for this function as explained in another section of this manual. The fill function can be used to clean up extraneous white space that may result from other file edits such as unwanted midline breaks and the REPLACE command.

The next examples demonstrate the fill function. The first example shows how the screen appears before executing the fill function and the select range that will go into the cut and paste buffer when the <DELETE LINE> key is pressed.

```

+-----+
|Line 1
|Line 2
|Line 3:first line of select range
|Line 4
|Line 5:last line of select range
|>>>>>SELECT LINE<<<<<<<
|Line 6
|>>>>buf_bof<<<<
|
|>>
|Msg:
|
+-----+

```

application

The next two examples show how the screen appears after pressing the <DELETE LINE> key and then pressing the <FILL> key from the cursor position shown. Note that the fill margins are the same as the workspace buffer margins in this example.

```
+-----+
|Line 1
|Line 2
|Line 6
|>>>>buf_bof<<<<
|
|
|
|>>
|Msg:                                application|
+-----+
```

```
+-----+
|Line 3:first line of select range Line 4 Line 5:last line
|of select range
|Line 6
|>>>>buf_bof<<<<
|
|
|
|>>
|Msg:                                application|
+-----+
```

3.8 Quitting

The <QUIT> key either ends the current edit session or cancels command mode and returns you to edit mode.

To incorporate your edits into an existing file or to create a new file from the workspace buffer, you must execute the SAVE command as explained in another section of this manual. If the workspace buffer has been modified and you press the <QUIT> key before saving, the message "Buffer has been changed, use <quit> to really quit". You can press the <QUIT> key again

to exit without saving or press any other function key to return to edit mode.

3.9 Executing Command Mode Commands

The `<COMMAND>` key is used to switch to command mode. The cursor then moves to the command line prompt (`>>`). At this point, if you are familiar with the command mode commands, you can enter the desired command with its operands and press the `<ENTER>` key. Operands are discussed in the sections explaining each command.

```
+-----+
|Line 1
|Line 2
|Line 3
|Line 4
|Line 5
|Line 6<COMMAND>
|>>>>buf_bof<<<<
|
|
|>> save curfile<ENTER>
|Msg:                                     application|
+-----+
```

Command names may be abbreviated as long as enough characters are entered to keep them unique (e.g., "se" may be entered for search). Each command or operand must be separated by a blank. Operands which contain a blank such as search and replace strings must be enclosed in double quotes ("). Examples of this will be shown for the specific commands this pertains to.

When a command has completed successfully, you automatically return to edit mode and the cursor will be in the same text position it was in when you first pressed the `<COMMAND>` key.

You can also execute command mode commands from the Command Mode Menu. This menu is displayed by pressing the `<HELP>` key from the command line prompt (e.g., `>> <HELP>`) or by leaving the command line blank or blank filling the command line and pressing the `<ENTER>` key. The next example shows the Command Mode Menu.

```
+-----+
|               |
|   Command List   |
|               |
|   Search:  _     |
|  Replace:  _     |
|    Load:         |
|    Save:         |
|   Clear:         |
|  Margins:         |
|  Repeat:         |
|   Buffer:         |
|               |
|   >>           |
|Msg:                                     application|
+-----+
```

To execute a command from the menu, move the cursor to the field with the desired command, enter the required operands, and press the <ENTER> key. The next example shows this.

```
+-----+
|               |
|   Command List   |
|               |
|   Search:         |
|  Replace:         |
|    Load:         |
|    Save: curfile<ENTER>|
|   Clear:         |
|  Margins:         |
|  Repeat:         |
|   Buffer:         |
|               |
|   >>           |
|Msg:                                     application|
+-----+
```

If you put the cursor on the desired command field and press the <HELP> key, prompts for the required operands appear below the menu. This appears on your screen as shown in the next example.

```
+-----+
|               Command List               |
|                                           |
| Search:                                     |
| Replace:                                    |
| Load:                                     |
| Save: <HELP>                               |
| Clear:                                     |
| Margins:                                   |
| Repeat:                                    |
| Buffer:                                     |
|                                           |
| Save a File                               |
|                                           |
| Enter File Name: _                       |
|                                           |
| >>                                        |
|Msg:                                     application|
+-----+
```

3.10 Searching for Text

The SEARCH command allows you to find the first occurrence of a given text string and positions the cursor at the first character of the string in the workspace buffer. If the SEARCH command is followed by a minus (e.g., search "text string" -), the search will be done backwards from the current cursor position towards the top of the file. Otherwise, the search will be done from the current cursor position towards the end of the file. The text string you are searching for can be a maximum of 40 characters long and must be enclosed in double quotes if it contains spaces. If the string is not found, the message "String not found" is displayed in the message line and you remain in the current command mode.

3.10.1 SEARCH Operands

The next example shows how the prompts for the SEARCH command operands appear on your screen. The operands entered the last time the search command was executed will appear as the current values for these fields.

3.11 Replacing One Text String with Another

The REPLACE command replaces a given "old string" with the specified "new string". The search for the old string always begins at the current cursor position.

3.11.1 Specifying the Scope

You have several options for specifying the scope of the search and replace. If a star (*) is appended to the command (e.g., replace "old string" "new string" *), the replacements will be made on all occurrences from the cursor to the end of the buffer. If a period is appended to the command, the replacements will be made on all occurrences from the cursor to the end of the line. Note that a star and a period may not be used in the same command. If neither of these options is specified, "new string" will replace the first occurrence of "old string" to the right of the current cursor position. If a minus (-) is appended to one of the above commands (e.g., replace "old string" "new string" . -), the replacements will be performed backwards. This means from the cursor to the top of the file, from the beginning of the line to the cursor, or the first occurrence to the left of the cursor respectively.

If you define a select range replacements will be performed on all occurrences of the "old string" in this range. The section, "Selecting Text to be Moved" explains defining a select range.

If the "old string" is not found, a message is displayed and you remain in command mode.

The cursor position after the REPLACE command is executed depends on the scope option that was specified. For commands appended with no option or a period, the cursor is positioned just after the last text replacement. For commands appended with a star, the cursor is positioned at the bottom of the file or at the end of the select range if that form of the command was used.

3.11.2 REPLACE Operands

The next example shows how the prompts for the REPLACE command operands appear on your screen.

```
+-----+
|           Command List           |
|                                     |
| Search:                            |
| Replace: <HELP>                     |
|   Load:                           |
|   Save:                              |
|   Clear:                             |
| Margins:                            |
| Repeat:                              |
| Buffer:                              |
|                                     |
|           Replace One String with Another           |
|                                     |
| Enter Old String: -                 |
| Enter New String: -                 |
|   Option:                           |
|   Direction:                        |
|                                     |
| >>                                     |
|Msg:                                     application|
+-----+
```

The command line format for this command is:

```
>> replace "old string" "new string" opt dir<ENTER>
```

where "opt" is the scope option and "dir" is the replacement direction (blank or -).

3.11.3 Considerations

A line that is longer after the replacement is wrapped and a shorter line is padded on the right with blanks. The extraneous whitespace this may cause can be eliminated by selecting the appropriate range of text and using the fill function.

A replace is not permitted on the bottom of file line or any following line. An attempt to do so will abort the command and a warning message is issued.

If you position the cursor at the REPLACE command on the command mode menu and press the <ENTER> key without entering any operands, the command will be reexecuted using the operands previously entered.

3.12 Loading a File for Editing

The LOAD command inserts a file into the workspace buffer just before the line containing the current cursor position. If you execute this command while there is text in the buffer, the loaded file becomes part of the current workspace buffer.

You can also use this command to start editing a new file as a separate entity without ending your current editing session. To do this you clear the workspace buffer using the CLEAR command before executing the LOAD command. You may also want to save the file currently in the workspace buffer before executing the CLEAR command. When you initially access the Text Editor, you can use the LOAD command without the SAVE and CLEAR commands to insert an existing file you want to edit into the buffer. You must press the <FIRST PAGE> key to display the first n lines of the file.

3.12.1 LOAD Operand

The next example shows how the prompt for the LOAD command operand appears on your screen.

```
+-----+
|           Command List           |
|                                     |
| Search:                           |
| Replace:                           |
|   Load: <HELP>                     |
|   Save:                             |
|   Clear:                            |
| Margins:                           |
| Repeat:                             |
| Buffer:                              |
|                                     |
| Load a File                         |
|                                     |
| Enter File Name: _                 |
|                                     |
| >>                                  |
|Msg:                                                                           |
+-----+ application
```

The command line format for this command is:

```
>> load filename<ENTER>
```

3.12.2 Considerations

The LOAD command cannot be executed if the current cursor position is below the bottom of file mark. An attempt to do so will abort the command and a warning message will be issued.

Failure of the I/O operations open file, close file, and read a record and the insert line function will also cause the load command to abort and a warning message will be issued.

The name of the loaded file does not become the default save name. A file name must be specified when the workspace buffer is saved or you can use the BUFFER command to set or change the default save name as explained in another section of this manual.

3.13 Saving the Workspace Buffer

The SAVE command writes the contents of the workspace buffer to the file name specified. If no file name is given, the buffer is written to the current buffer name. The current buffer name is either the name of the file specified when the TE was accessed or the buffer name specified by executing the BUFFER command.

3.13.1 SAVE Operand

The next example shows how the prompt for the SAVE command operand appears on your screen.

```
+-----+
|           Command List           |
|                                     |
|   Search:                          |
|   Replace:                          |
|     Load:                          |
|     Save: <HELP>                    |
|     Clear:                          |
|   Margins:                          |
|     Repeat:                          |
|     Buffer:                          |
|                                     |
|   Save a File                       |
|                                     |
|   Enter File Name: _                |
|   >>                                |
|Msg:                                     application|
+-----+
```

The command line format for this command is:

>> save filename<ENTER>

When the SAVE command is complete the cursor returns to position it was in before you went into command mode.

3.13.2 Considerations

If the I/O operations open file, close file and write a record fail, the save command is aborted. The state of the file on such a failure is system dependent.

If no file name is supplied and there is no buffer name (default save name) when the SAVE command is executed, a message is displayed and one remains in the current command mode. This condition results if the file was brought into the TE using the LOAD command and the workspace buffer was empty.

3.14 Clearing the Workspace Buffer

The CLEAR command deletes all lines between the top of file and bottom of file marks. The lines become the current contents of the cut and paste buffer. The CLEAR command can be undone using the paste function and naming the buffer.

This command is useful if you want to edit another file as a separate entity during your current edit session. To do this

you would first save the current workspace buffer and then execute the CLEAR command. Then execute the LOAD command to insert the new file into the workspace buffer and change the default save name using the BUFFER command.

The CLEAR command has no operands or considerations.

3.15 Setting the Fill Margins

The MARGINS command is used to set the left and right margins that will be used when the fill function is executed. The margin values must satisfy the constraint:

$$0 \leq \text{"left margin"} < \text{"right margin"} \leq \text{display width} + 1$$

This means that if the left margin is 9 and the right margin is 70, each line printed can have up to 60 characters in columns 10 through 69. The default fill margin values are the same as the margins that control the width of the display page. For a VT100 terminal, the display width is 79 so the default fill margin values are 0 and 80.

3.15.1 MARGINS Operands

The next example shows how the prompts for the MARGINS command operands appear on your screen.

```
+-----+
|           Command List           |
|                                     |
| Search:                           |
| Replace:                           |
|   Load:                           |
|   Save:                             |
|   Clear:                            |
| Margins: (HELP)                    |
| Repeat:                             |
| Buffer:                              |
|                                     |
| Set Fill Margins                   |
|                                     |
| Left Margin: _                     |
| Right Margin: _                    |
| >>                                 |
| Msg:                               |                               application|
+-----+
```

The command line format for this command is:

```
>> margins lm rm<ENTER>
```

where "lm" is the left fill margin value and "rm" is the right fill margin value.

3.15.2 Considerations

The margin values set with this command are only used by the fill function. The workspace buffer margins are permanently set based on terminal type. For example, on a VT100, the margins are 0 and 80.

3.15.3 Example Showing Affects of Setting Fill Margins

The next examples demonstrate the fill function when the fill margins have been set to 0,28. The first example shows how the screen appears before the fill function and the select range that will go into the cut and paste buffer when the <DELETE LINE> key is pressed from the cursor position shown.

```
+-----+
|Line 1 of the workspace buffer. |
|Line 2 of the workspace buffer. |
|Line 3: first line of select range |
|Line 4 of the workspace buffer. |
|Line 5: last line of select range |
|>>>>>SELECT LINE<<<<<<<< |
|Line 6 of the workspace buffer. |
|>>>>buf_bof<<<< |
| |
|>> |
|Msg: | application|
+-----+
```

The next two examples show how the screen appears after pressing the <DELETE LINE> key and then pressing the <FILL> key from the cursor position shown.

```
+-----+
|Line 1 of the workspace buffer.
|Line 2 of the workspace buffer.
|Line 6 of the workspace buffer.
|>>>>buf_bof<<<<
|
|
|
|>>
|Msg:                                     application|
+-----+
```

```
+-----+
|Line 1 of the workspace buffer.
|Line 2 of the workspace buffer.
|Line 3: first line of select
|range Line 4 Line 5: last
|line of select range
|Line 6 of the workspace buffer.
|>>>>buf_bof<<<<
|
|
|
|>>
|Msg:                                     application|
+-----+
```

3.16 Repeating a Command or Function Key

The REPEAT command allows you to specify that the next command executed or function key pressed will automatically be repeated a specified number of times. When you execute this command, the message "in repeat" is displayed in the message line. You do not return to edit mode until after you press the desired function key or execute the desired command. When the last repetition has completed, the message "operation complete" is displayed in the message line and then you return to edit mode. The function keys that are supported by the REPEAT command are <INSERT LINE>, <DELETE LINE>, <FIRST PAGE>, <LAST PAGE>, <PASTE>, <FILL>, <MIDLINE BREAK>, and <QUIT>. However, the <FIRST PAGE>, <LAST PAGE>, and <QUIT> functions will only be executed once. The cursor position after the the last repetition has completed depends on the function or command performed. Pressing the <QUIT> key essentially cancels the

repeat command and the cursor is returned to the position it was in before going into command mode.

3.16.1 REPEAT Operands

The next example shows how the prompt for the REPEAT command operand appears on your screen.

```
+-----+
|           Command List           |
|                                     |
|   Search:                          |
|   Replace:                          |
|     Load:                          |
|     Save:                           |
|   Clear:                            |
|   Margins:                          |
|   Repeat: <HELP>                    |
|   Buffer:                            |
|                                     |
|   Repeat Command/Function          |
|                                     |
|   Enter Repeat: _                  |
| >>                                  |
|Msg:                                application|
+-----+
```

The command line format for this command is:

```
>> Repeat n<ENTER>
```

where "n" is the number of times you want to repeat the command or function.

3.16.2 Considerations

The number of repetitions must be greater than zero. If it is not, the command or function is aborted and a warning message is issued.

3.17 Naming the Workspace Buffer

The BUFFER command allows you to specify a default file name for the SAVE command. This is the file the workspace buffer will be written to if you execute the SAVE command without giving a file name. This name overrides the default save name that was set if a file name was specified when the TE was accessed.

3.17.1 BUFFER Operand

The next example shows how the prompt for the BUFFER command operand appears on your screen.

```
+-----+
|           Command List           |
|                                   |
|   Search:                         |
|   Replace:                        |
|     Load:                         |
|     Save:                          |
|   Clear:                           |
| Margins:                           |
|   Repeat:                          |
|   Buffer: <HELP>                   |
|                                   |
|           Buffer Name              |
|                                   |
| Enter Buffer Name: _               |
| >>                                |
|Msg:                                application|
+-----+
```

The command line format for this command is:

```
>> buffer filename<ENTER>
```

The BUFFER command has no considerations.

APPENDIX A

FORM PROCESSOR (FP)/TEXT EDITOR (TE)
CALLABLE INTERFACE

The TE callable interface allows access to the TE file editing capabilities (as described in the TE DS and UM) from a program. The calling requirements are as follows:

```
void editci(file_name, changed)
    char file_name[30];
    int  *changed;
```

The filename is a thirty character string with the name of the file to be brought into the buffer for editing. It must be blank filled in all languages except C.

The changed flag is set to true if the buffer is subsequently modified and saved, else it is set to false.

Notes:

1. The FP procedure initfp must be called before the first call to editci.
2. Since the TE is essentially an FP application all messages are directed to the message line of the screen.
3. When linking in editci the usual FP and VTI linking requirements are to be used as well.
4. When running an application with editci the usual FP and VTI run time requirements hold.

END

8-87

DTIC